**Use Cases**

for

<SG Parking>

Version 1.0 approved

Prepared by <Kenny, Zhenkai, Swee Ngee, Isaac, lynn>

<No Turning Bac>

<03/02/2020>

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Date** | **Reason For Changes** | **Version** |
|  |  |  |  |
|  |  |  |  |

# **Guidance for Use Case Template**

Document each use case using the template shown in the Appendix. This section provides a description of each section in the use case template.

# **1.** **Use Case Identification**

## **1.1.** **Use Case ID**

Give each use case a unique numeric identifier, in hierarchical form: X.Y. Related use cases can be grouped in the hierarchy. Functional requirements can be traced back to a labeled use case.

## **1.2.** **Use Case Name**

State a concise, results-oriented name for the use case. These reflect the tasks the user needs to be able to accomplish using the system. Include an action verb and a noun. Some examples:

· View part number information.

· Manually mark hypertext source and establish link to target.

· Place an order for a CD with the updated software version.

## **1.3.** **Use Case History**

### **1.3.1** **Created By**

Supply the name of the person who initially documented this use case.

### **1.3.2** **Date Created**

Enter the date on which the use case was initially documented.

### **1.3.3** **Last Updated By**

Supply the name of the person who performed the most recent update to the use case description.

### **1.3.4** **Date Last Updated**

Enter the date on which the use case was most recently updated.

# **2.** **Use Case Definition**

## **2.1.** **Actor**

An actor is a person or other entity external to the software system being specified who interacts with the system and performs use cases to accomplish tasks. Different actors often correspond to different user classes, or roles, identified from the customer community that will use the product. Name the actor(s) that will be performing this use case.

## **2.2.** **Description**

Provide a brief description of the reason for and outcome of this use case, or a high-level description of the sequence of actions and the outcome of executing the use case.

## **2.3.** **Preconditions**

List any activities that must take place, or any conditions that must be true, before the use case can be started. Number each precondition. Examples:

1. User’s identity has been authenticated.

2. User’s computer has sufficient free memory available to launch task.

## **2.4.** **Postconditions**

Describe the state of the system at the conclusion of the use case execution. Number each postcondition. Examples:

1. Document contains only valid SGML tags.

2. Price of item in database has been updated with new value.

## **2.5.** **Priority**

Indicate the relative priority of implementing the functionality required to allow this use case to be executed. The priority scheme used must be the same as that used in the software requirements specification.

## **2.6.** **Frequency of Use**

Estimate the number of times this use case will be performed by the actors per some appropriate unit of time.

## **2.7.** **Flow of Events**

Provide a detailed description of the user actions and system responses that will take place during execution of the use case under normal, expected conditions. This dialog sequence will ultimately lead to accomplishing the goal stated in the use case name and description. This description may be written as an answer to the hypothetical question, “How do I <accomplish the task stated in the use case name>?” This is best done as a numbered list of actions performed by the actor, alternating with responses provided by the system.

## **2.8.** **Alternative Flows**

Document other, legitimate usage scenarios that can take place within this use case separately in this section. State the alternative course, and describe any differences in the sequence of steps that take place. Number each alternative course using the Use Case ID as a prefix, followed by “AC” to indicate “Alternative Course”. Example: X.Y.AC.1.

## **2.9.** **Exceptions**

Describe any anticipated error conditions that could occur during execution of the use case, and define how the system is to respond to those conditions. Also, describe how the system is to respond if the use case execution fails for some unanticipated reason. Number each exception using the Use Case ID as a prefix, followed by “EX” to indicate “Exception”. Example: X.Y.EX.1.

## **2.10.** **Includes**

List any other use cases that are included (“called”) by this use case. Common functionality that appears in multiple use cases can be split out into a separate use case that is included by the ones that need that common functionality.

## **2.11.** **Special Requirements**

Identify any additional requirements, such as nonfunctional requirements, for the use case that may need to be addressed during design or implementation. These may include performance requirements or other quality attributes.

## **2.12.** **Assumptions**

List any assumptions that were made in the analysis that led to accepting this use case into the product description and writing the use case description.

## **2.13.** **Notes and Issues**

List any additional comments about this use case or any remaining open issues or TBDs (To Be Determined) that must be resolved. Identify who will resolve each issue, the due date, and what the resolution ultimately is.

## **3.** **Use Case Description3**

## **3.1.** **Use cases**

* UC001 Customer create an account
* UC002 Customer log into application
* UC003 GPS notification
* UC004 Carpark Search (listing view/map view)
* UC005 Sorting (before/after search)
* UC006 Filtering (after search)
* UC007 Route Planning (from current location to parking lot)
* UC008 Route Planning (exiting car park from HDB and entering main road)
* UC009 Report a listed car park
* UC010 Access the help center
* UC011 Tutorial

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID: | UC001 | | |
| Use Case Name: | Customer create an account | | |
| Created By: | Kenny Lew | Last Updated By: | Kenny Lew |
| Date Created: | 1/02/2020 | Date Last Updated: | 1/02/2020 |
| Actor: | Application User (initiating actor) | | |
| Description: | The user signs up with the application | | |
| Preconditions: | 1. The user has downloaded the application  2. The user has a personal email | | |
| Postconditions: | User has successfully created an account with the application | | |
| Priority: |  | | |
| Frequency of Use: | Edge use case | | |
| Flow of Events: | 1. The user click on sign up in the application home page  2. The system navigates to the sign up page and choose sign up as customer.  3. The system requests the user to sign up with email and the system displays the button “Already Have an Account? Sign In”  4. The user will enter their personal email and set the account password twice.  5. The system sends a verification code to the entered email.  6. The user inputs the verification code.  7. The system reports to the user that the sign up is successful.  8. The user is redirected to the main application page | | |
| Alternative Flows: | AF-S4: If the user inputs an invalid email address to sign up.   1. The system displays a message “Invalid email address”. 2. The system returns to the beginning of step 4.     AF-S4: If the user inputs an email address that has already been used for signing up.   1. The system displays a message “The email has been used for signing up”. 2. The system returns to the beginning of step 4.     AF-S4: If the user input two different passwords when setting the password.   1. The system displays message “Passwords don’t match, please set again” 2. The system returns to the beginning of step 4.     AF-S4: If the user inputs the wrong verification code.   1. The system displays message “Wrong Verification Code” 2. The system returns to the beginning of step 4 and asks the user to choose whether to resend the verification code. | | |
| Exceptions: | EX1: The user click the cancel button during the signing up process.   1. The sign up process terminates and display a message “Quit sign up”   EX2: The user already have an account to sign in, and click the “Sign in” in the step 4   1. The sign up process terminates and, 2. The system navigates to sign in page. | | |
| Includes: |  | | |
| Special Requirements: |  | | |
| Assumptions: |  | | |
| Notes and Issues: |  | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID: | UC002 | | |
| Use Case Name: | Customer log in to application | | |
| Created By: | Kenny Lew | Last Updated By: | Kenny Lew |
| Date Created: | 1/02/2020 | Date Last Updated: | 1/02/2020 |
| Actor: | Application User (initiating actor) | | |
| Description: | The user logs in to the application | | |
| Preconditions: | 1. The user has downloaded the application  2. The user has an account the application | | |
| Postconditions: | User has successfully logged in to the application | | |
| Priority: |  | | |
| Frequency of Use: | Necessary case | | |
| Flow of Events: | 1. The user click on the log in button in the application home page  2. The system navigates to the log in page and choose sign up as customer.  3. The system requests the user to sign in with email password the system displays the message “No account? Sign up here”  4. The user will enter their personal email and their registered password.  5. The system verifies whether the email and password are correct  6. Once verified, the system redirects the user to the home page as a logged-in user  7. The user can now use the application | | |
| Alternative Flows: | AF-S4: If the user inputs an invalid email address to log in.   1. The system displays a message “Email address or password incorrect”. 2. The system returns to beginning of step 5.   AF-S4: If the user inputs the wrong password for the email address provided.   1. The system displays a message “Email address or password incorrect” 2. The system returns to the beginning of step 5 | | |
| Exceptions: | EX1: The user click the cancel button during the log in process.   1. The log in process terminates and redirects the user back to the home page   EX2: The user does not have an account to log in, and click the “Sign up” in the step 3   1. The sign up process terminates and, 2. The system navigates to sign up page. | | |
| Includes: |  | | |
| Special Requirements: |  | | |
| Assumptions: |  | | |
| Notes and Issues: |  | | |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Use Case ID: | UC003 | | | | | |
| Use Case Name: | GPS notification | | | | | |
| Created By: | zhenkai | | | Last Updated By: | | zhenkai |
| Date Created: | 02/02/2020 | | | Date Last Updated: | | 02/02/2020 |
| Actor: | Application User (initiating actor) | | | | | |
| Description: | The app will display a pop-up dialog that prompts user to on their GPS function. | | | | | |
| Preconditions: | 1. The user has downloaded the application, and  2. The GPS function is turned off | | | | | |
| Postconditions: | GPS turned on | | | | | |
| Priority: |  | | | | | |
| Frequency of Use: | Daily use | | | | | |
| Flow of Events: | 1. The user activates the application  2. The application detects that GPS function is not on  3. The application detects that GPS function is not on and display a pop-up dialog stating: “Please turn on the GPS before you can proceed”. A “OK” button will also be displayed in the pop-up dialog  4. The user proceed to the setting or the quick toolbar to on the GPS  5. The user return to the application and click on the “OK” button.  6. The dialog disappear when the GPS is on and the “OK” button is pressed | | | | | |
| Alternative Flows: | AF-S4: If the user is on airplane mode.   1. The GPS function will never be on. 2. There will be no notification to inform user that he/she is in airplane mode and GPS is disabled   AF-S6: If the user clicks on the “OK” button without turning the GPS function on.   1. The pop-up dialog displayed by the application will remain 2. The application will stay at step 3 until user quit the application or turn the GPS on, | | | | | |
| Exceptions: |  | | | | | |
| Includes: |  | | | | | |
| Special Requirements: |  | | | | | |
| Assumptions: | All android smartphones have GPS function in it. | | | | | |
| Notes and Issues: |  | | | | | |
| Use Case ID: | | UC004 | | | | | |
| Use Case Name: | | Carpark search (listing view/map view) | | | | | |
| Created By: | | zhenkai | Last Updated By: | | zhenkai | | |
| Date Created: | | 02/02/2020 | Date Last Updated: | | 02/02/2020 | | |
| Actor: | | Application User (initiating actor) | | | | | |
| Description: | | The application will have a search bar that allows user to perform search function. | | | | | |
| Preconditions: | | 1. The user has downloaded the application, and  2. The GPS function is turned ON | | | | | |
| Postconditions: | | Display the carpark information | | | | | |
| Priority: | |  | | | | | |
| Frequency of Use: | | Daily use | | | | | |
| Flow of Events: | | 1. The user will key in the information in the search bar on the top of the application. There exists the "listing view" tab and "map view" tab. The user will stay in the default "map view" tab.  2. The search bar will have annotation of “carpark location”  3. The user will fill in the search bar before clicking the “Search” button.  4. The application will receive the information from the user and access the government API for results that matches user’s input.  5. The application will then display a location pin that best matches the carpark location information received.  6. The map will display the nearest carparks from the pin using the default search radius from the results received from the API base, and will also display the carparks in list formats in order of relevance below the map  7. The user will then select the carpark he/she wants from the map.  8. Application will bring the user to another display page where for details of the carpark will be displayed. E.g. carpark name, price of the carpark in SGD per hour, Distance of the carpark from the user’s current location, etc. | | | | | |
| Alternative Flows: | | AF-S1: User selects "listing view" tab and proceed to search for carpark information.  1. Step 2-4 will remain the same.  2. Step 5: Pin location on the map will not be displayed.  3. Step 6: Application will display the carpark information in a listed view (sorted/unsorted) below the search bar.  4. Step 7: User will select the carpark he/she wants from the list  5. Step 8 will remain the same  AF-S5: If no relevant information of the carpark can be found.  1. The application will display “No search result found”  AF-S6: If the user decides to check for another carpark information through step 1.   1. The previous search result will remain while user key in the new search input 2. Application will then repeat step 4 & 5 and return to step 5 with the new result. | | | | | |
| Exceptions: | | EX1: User at any point of time can click on the built-in return button in any android smartphone to return to the previous step. Input in the current page will not be stored.  EX2: If the user did not fill in the search bars and press on the “Search button”, application will display message “Please fill in the search bars before proceeding”. The same message will be displayed repeatedly until the user has an input. | | | | | |
| Includes: | |  | | | | | |
| Special Requirements: | |  | | | | | |
| Assumptions: | |  | | | | | |
| Notes and Issues: | |  | | | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID: | UC005 | | |
| Use Case Name: | Sorting (before/after search) | | |
| Created By: | zhenkai | Last Updated By: | zhenkai |
| Date Created: | 02/02/2020 | Date Last Updated: | 02/02/2020 |
| Actor: | Application User (initiating actor) | | |
| Description: | The application will have a sorting button on the right of the search bar that allows user to perform sorting function. | | |
| Preconditions: | 1. The user has downloaded the application, and  2. The GPS function is turned ON  3. The user has login | | |
| Postconditions: | User can select one of following: alphabetical order/price/distance/number of unoccupied slots. | | |
| Priority: |  | | |
| Frequency of Use: | Daily use | | |
| Flow of Events: | 1. The user will press on the “Sort” button.  2. The application will display pop-up with a total of 4 options, the user can only select one of the options at all time. Either one of the following: alphabetical order/price/distance/number of unoccupied slots. There will also be a “Apply button”  3. The user will select one of those options and click on the “Apply button”  4. The pop-up window will be close and a text message that display: “Applied successfully” will be shown for 2 seconds | | |
| Alternative Flows: | AF-S4: If the user has already performed a search function and decide to use the sort function   1. The previous search result will be display but in the sorted order | | |
| Exceptions: | EX1: User at any point of time can click on the built-in return button in any android smartphone to return to the previous step. Input in the current page will not be stored. | | |
| Includes: |  | | |
| Special Requirements: |  | | |
| Assumptions: |  | | |
| Notes and Issues: |  | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID: | UC006 | | |
| Use Case Name: | Filtering (after search) | | |
| Created By: | zhenkai | Last Updated By: | zhenkai |
| Date Created: | 02/02/2020 | Date Last Updated: | 02/02/2020 |
| Actor: | Application User (initiating actor) | | |
| Description: | The application will have a filter button on the right of the sorting button that allows user to perform filtering function. | | |
| Preconditions: | 1. The user has downloaded the application, and  2. The GPS function is turned ON | | |
| Postconditions: | Display the filtered search result. | | |
| Priority: |  | | |
| Frequency of Use: | Daily use | | |
| Flow of Events: | 1. The user will press on the “Filter” button.  2. The application will display pop-up with a total of 3 sliders, the user can choose to filter the result based on these sliders. Each slider will have an annotation of Maximum distance, Maximum price, Minimum carpark availability (%). There are 2 tabs, the "indoor" and "outdoor" tab, and users can select them to filter carpark information. There will also be a “Apply button”  3. Users will then drag the slider to a desired amount and select the "indoor" tab. Users can only select one tab at all times. User will then click on the “Apply” button.  4. The pop-up window will be closed and a text message that display: “Applied successfully” will be shown.  5. Even if there is no input, “Applied successfully” will be shown.  6. The application will now show the filtered result. | | |
| Alternative Flows: | AF-S3: If the user click on the "indoor" tab again.   1. The "indoor" tab will not be highlighted and the search result will return both indoor and outdoor carpark result   AF-S3: If the user click on the "outdoor" tab.   1. The "indoor" tab will not be highlighted and the "outdoor" tab will be highlighted   AF-S4: If the user has already performed a search function and decide to use the filter function   1. The previous search result will be filtered and display. | | |
| Exceptions: | EX1: User at any point of time can click on the built-in return button in any android smartphone to return to the previous step. Input in the current page will not be stored.  EX2: None of the search result fits the user’s filter requirement. The application will display “No search result found”. | | |
| Includes: |  | | |
| Special Requirements: |  | | |
| Assumptions: |  | | |
| Notes and Issues: |  | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID: | UC007 | | |
| Use Case Name: | Route Planning (From current location to parking lot) | | |
| Created By: | Swee Ngee | Last Updated By: | Swee Ngee |
| Date Created: | 2/2/2020 | Date Last Updated: | 3/2/2020 |

|  |  |
| --- | --- |
| Actor: | Application User (Initiating Actor) |
| Description: | System suggests optimal route towards parking lot that user had chosen |
| Preconditions: | User selected the parking lot to go to |
| Postconditions: | System generates multiple routes for user to choose with order, highest ranking being the fastest |
| Priority: |  |
| Frequency of Use: | Daily use scenario |
| Flow of Events: | 1. The pop-up window will be close and a text message that display: “Applied successfully” will be shown for 2 seconds   2.The system calls Google Map API, Government Car Park Availability API to generate feasible routes plans  3.The top three fastest route will be shown to the user  4.User have the freedom to choose the three routes, the highest-ranking route being the default route |
| Alternative Flows: | AF-S4: User wants to avoid tolls   1. User to click on the option “avoid tolls” 2. Repeat steps 2-4 |
| Exceptions: | EX-S2: System cannot generate route if all roads leading to the parking lot is closed/not operational, or system could not call the API.  1. The system displays service unavailable  2. User returns to choosing carpark page |
| Includes: |  |
| Special Requirements: |  |
| Assumptions: |  |
| Notes and Issues: |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID: | UC008 | | |
| Use Case Name: | Route Planning (Exiting carpark from HDB and entering main road) | | |
| Created By: | Swee Ngee | Last Updated By: | Swee Ngee |
| Date Created: | 2/2/2020 | Date Last Updated: | 3/2/2020 |

|  |  |
| --- | --- |
| Actor: | Application User (Initiating actor) |
| Description: | System navigate shortest route towards the main road from the HDB carpark |
| Preconditions: | 1. User has completed UC003 2. User is still using app and pressed navigate towards main road option |
| Postconditions: | System generates shortest route to the exit |
| Priority: |  |
| Frequency of Use: | Daily use scenario |
| Flow of Events: | 1. The system calls Google Map API, to check if user is within vicinity of HDB carpark. 2. System prompts user to request to offer help for user to navigate out towards the main road 3. User agrees and press the “Yes” button on the interface 4. The system calls Google Map API, Government HDB Carpark Information API to generate shortest route |
| Alternative Flows: | AF-S3: User does not want any help  3. User press “No” button on interface  4. User goes back to main page of application |
| Exceptions: | EX-S1: System cannot generate route if user is not parking in the HDB vicinity  1. The system will not prompt user to offer help  2. Application goes back to main page |
| Includes: |  |
| Special Requirements: |  |
| Assumptions: |  |
| Notes and Issues: |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID: | UC009 | | |
| Use Case Name: | Report a listed carpark | | |
| Created By: | Lynn Htet Aung | Last Updated By: |  |
| Date Created: | 03/02/2019 | Date Last Updated: |  |
| Actor: | Application User (initiating actor) | | |
| Description: | The app user flags false/inaccurate information for the carpark selected | | |
| Preconditions: | 1. The user has logged successfully into the application using a valid account  2. The user has selected a carpark from the map or from search results | | |
| Postconditions: | The application will register the flag in the database for the dev to take further actions | | |
| Priority: |  | | |
| Frequency of Use: | Edge use case (seldom used) | | |
| Flow of Events: | 1. The user proceeds to click the ‘Report’ button 2. The system navigates to the reporting page 3. The user can check the attributes/statuses of the car park to flag as inaccurate 4. The system will prompt the user to enter/ select updated information for each checked attribute 5. The user proceeds to click the ‘Update’ button 6. The system displays a pop-up dialog that displays “Are you sure to report this carpark?” and prompts a confirmation from the user. 7. The system indicates that the user flag has been reported to the developer | | |
| Alternative Flows: | AF-S4: If the user does not input any suggested value for each flagged attribute   1. The system displays a message “Are you sure you do not want to suggest updated information for the carpark?”. 2. The system asks for user confirmation. 3. If the user clicks “No”, system returns to step 3 | | |
| Exceptions: | EX1: The user click the cross button during the reporting process.   1. The reporting process terminates and display a message “Quit fault reporting”   EX2: The user inputs an invalid parameter when suggesting alternate attributes of the carpark   1. The system highlights the invalid input in red and prompts the user to enter a valid input | | |
| Includes: |  | | |
| Special Requirements: |  | | |
| Assumptions: |  | | |
| Notes and Issues: |  | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID: | UC010 | | |
| Use Case Name: | Access the help center | | |
| Created By: | Lynn Htet Aung | Last Updated By: |  |
| Date Created: | 03/02/2019 | Date Last Updated: |  |
| Actor: | Application User (initiating actor) | | |
| Description: | The app user gets help from the help center on using the application | | |
| Preconditions: | 1. The user has opened the application | | |
| Postconditions: |  | | |
| Priority: | High | | |
| Frequency of Use: | Edge use case (seldom used) | | |
| Flow of Events: | 1. The user proceeds to click the ‘Help center’ button 2. The system navigates to the help center page 3. The user can select from a variety of Frequently Asked Questions 4. The system will navigate to descriptions of solution for each question 5. The user can choose to ‘Contact the development team’ 6. The system asks for name, contact number, and description of problem 7. The system indicates that the user query has been forwarded with high priority to the developer | | |
| Alternative Flows: | AF-S4: If the user chooses to contact the dev team through the team email.  1. The app copies the email address to the device’s clipboard when the user presses ‘Email the devs’  2. The system displays message “Email address of the dev team has been copied to clipboard” | | |
| Exceptions: | EX1: The user click the cross button from the help center.  1. The system process terminates and display a message “Quit help center”  EX2: The user inputs an invalid parameter when contacting the dev team  1. The system highlights the invalid input in red and re-prompts the user to enter a valid input  EX3: The user skips any inputs when contacting the dev team  1. The system re-prompts the user to input all necessary details during the query | | |
| Includes: |  | | |
| Special Requirements: |  | | |
| Assumptions: |  | | |
| Notes and Issues: |  | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID: | UC011 | | |
| Use Case Name: | Tutorial | | |
| Created By: | Swee Ngee | Last Updated By: | Swee Ngee |
| Date Created: | 4/02/2020 | Date Last Updated: | 4/02/2020 |
| Actor: | Application User (initiating actor) | | |
| Description: | Process of going through the tutorial to understand how the application works | | |
| Preconditions: | 1. The user has logged in  2. The user is a first-time user | | |
| Postconditions: | User will be able to understand and navigate through the application | | |
| Priority: |  | | |
| Frequency of Use: | Necessary case | | |
| Flow of Events: | 1. The system picks up that user is using the application for the very first-time based on the timestamp of user taken from the database 2. System redirects user to the tutorial page 3. User undergoes tutorial 4. User finishes tutorial 5. Redirects user to main application page | | |
| Alternative Flows: | AF-S3: If the user does not want to have tutorial  1. User clicks “exit” button during the tutorial  2. Redirects user to main application page    AF-S1: If the user is not first-time user but still wants to have tutorial  1. User to click on the “help” button, then “tutorial” button  2. Continue from step 3-5 | | |
| Exceptions: | EX-S2: User exits the application while have tutorial  1. System will cache the tutorial process  2. User redirected back to the tutorial page | | |
| Includes: |  | | |
| Special Requirements: |  | | |
| Assumptions: |  | | |